Computer Physics Communications 202 (2016) 233–261

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

An object-oriented implementation of a parallel Monte Carlo code for radiation transport[☆]

Pedro Duarte Santos, Andrea Lani [*]

Von Karman Institute for Fluid Dynamics, Waterloosesteenweg 72, 1640, Sint Genesius Rode, Belgium

ARTICLE INFO

Article history:
Received 21 May 2015
Received in revised form
9 December 2015
Accepted 19 December 2015
Available online 8 January 2016

Keywords:
Radiation
Monte-Carlo
FIRE II
Re-entry
Particle-tracking
Hypersonics
Aerothermodynamics

ABSTRACT

This paper describes the main features of a state-of-the-art Monte Carlo solver for radiation transport which has been implemented within COOLFluiD, a world-class open source object-oriented platform for scientific simulations. The Monte Carlo code makes use of efficient ray tracing algorithms (for 2D, axisymmetric and 3D arbitrary unstructured meshes) which are described in detail. The solver accuracy is first verified in testcases for which analytical solutions are available, then validated for a space re-entry flight experiment (i.e. FIRE II) for which comparisons against both experiments and reference numerical solutions are provided. Through the flexible design of the physical models, ray tracing and parallelization strategy (fully reusing the mesh decomposition inherited by the fluid simulator), the implementation was made efficient and reusable.

Program summary

Program title: COOLFluiD-MC

Catalogue identifier: AEZG_v1_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AEZG_v1_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: GNU General Public License, version 3

No. of lines in distributed program, including test data, etc.: 1990165

No. of bytes in distributed program, including test data, etc.: 149533288

Distribution format: tar.gz

Programming language: C++.

Computer: From desktops to large HPC distributed systems.

Operating system: Mac OS X, Linux.

Has the code been vectorized or parallelized?: Parallelized through MPI

RAM: Depending on the problem size from a few Megabytes to several Gigabytes.

Classification: 21.2.

External routines: MPI, boost, PETSc, ParMETIS, cmake

Nature of problem: Radiative processes play a fundamental role in countless science and engineering contexts, including combustion, astrophysics, atmospheric space re-entry, experiments in plasma facilities (e.g. shock tubes, arc jets). The problem we are interested in is the computation of radiative heat transfer on arbitrarily complex geometries, in particular to characterize thermal loads acting on the surface of space vehicles.

Solution method: Our C++ code implements a flexible and efficient Monte Carlo algorithm making use of state-of-the-art ray tracing techniques designed for and validated on arbitrary unstructured meshes. The code is organized as a collection of dynamically linked libraries connected to the computational

---

[☆] This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (http://www.sciencedirect.com/science/journal/00104655).

[*] Corresponding author.
E-mail addresses: pedro.duarte.santos@vki.ac.be (P.D. Santos), lani@vki.ac.be (A. Lani).

http://dx.doi.org/10.1016/j.cpc.2015.12.017

kernel of the open source COOLFluiD platform. The Monte Carlo method is parallelized through mesh decomposition, while reusing the same partitioned mesh (and associated data structures) on which fluid dynamics equations are solved.

Unusual features: The radiation transfer code of COOLFluiD-MC offers a unique combination of ray tracing algorithms suitable for handling 2D, 2D axisymmetric and 3D simulations on arbitrary unstructured grids. The parallelization strategy is scalable and based on domain decomposition. The overall object-oriented design allows for easy integration of new models and algorithms which can be plugged in dynamically through self-registration techniques.

Additional comments: The code is part of a much larger set of COOLFluiD libraries, which are fully available on Github at https://github.com/andrealani/COOLFluiD and include multiple numerical solvers, physico-chemical models, processing algorithms and interfaces to third party scientific software. The CPC distribution also includes a state-of-the-art fully implicit Finite Volume solver for Euler and Navier–Stokes systems which is used to produce the flow solution for the radiative transport algorithm described in the paper. The aerothermodynamic solver, which is used for the final testcase (FIREII) in our paper, is based on the same core Finite Volume modules but on different physico-chemical and radiation models. The latter cannot be made available through the CPC library since they rely on third party libraries (MUTATION version 2.0 and PARADE version 3.1) which cannot be distributed due to NDA.
!!!!! The distribution file for this program is over 149 Mbytes and therefore is not delivered directly when download or Email is requested. Instead a html file giving details of how the program can be obtained is sent. !!!!!

Running time: Depending on the size of the computational grid, if it is 2D or 3D, on the number of processors used, the running time can vary from a few seconds up to hours or days. The testcases that are included with this version of the code, all using relatively small 2D or 3D meshes, run in 1–10 s each on 8 CPU-cores. The user is, however, totally free to run the same cases on a different number of cores (more or less, according to his/her convenience).

0. Introduction

Radiation plays a fundamental role in high-temperature and high-enthalpy processes. Combustion, solar chromosphere physics and space re-entry are just a few application examples where radiative processes are as important as fluid dynamics. When compared to convection and conduction, the continuum assumption in radiation heat transfer cannot be carried out, as the mean free path of the photons can be arbitrarily large [1]. Additionally, the radiative properties can also vary sharply in direction, space and wavelength. Radiation modeling requires the solution of the Radiative Transfer Equation (RTE), which is Lagrangian in nature. Due to the fundamentally different mathematical properties, the numerical tools developed for the Partial Differential Equations (PDE) governing the evolution of continuum flows cannot be reused for radiation modeling.

Different algorithms exist for solving the Radiative Transfer Equation (RTE), with different levels of accuracy and computational effort. Modest [1] and Howell [2] identify the four most relevant methods for aerothermodynamic applications, here ordered with increasing computational cost and accuracy. First, the tangent slab approximation, reducing the problem to a one-dimensional radiation heat transfer formulation, with known analytical solution. Second, the Spherical Harmonics (SH) method [3]. This method rewrites the RTE into an infinite series of terms in distance and direction. The series is then truncated into an easily solved set of equations. Unfortunately, this method is relatively inaccurate for low orders P0, P1 and, for high orders, no generic solution exists. The third is the Discrete Ordinances Method (DOM) [4]. It approximates the directional-dependent integrals into a set of numerically solved quadratures with constant properties. For exact descriptions of the radiative field, this method converges to the exact radiative heat flux with the increase of the number of spectral, directional and spacial quadratures. Non the less, for low number of quadratures, this method is known to cause ray effects [5], that will reduce the quality of the solution. The last method is Monte Carlo [6]. The energy of the medium is divided into a integer number of particles. Their properties, like wavelength, direction or position are randomly assigned, biasing this choice for the most relevant ones. The method will approach exactness when the number of particles increases. Some authors also propose hybrid formulations, like the work of Feldick [7], combining the fast P–N method with the accurate Monte Carlo algorithm to obtain higher computational efficiencies. Compared with the other methods discussed, Monte Carlo does not introduce any approximations and is straightforward to implement and parallelize with a domain decomposition, where each processor keeps only a part of the mesh. The treatment of scattering and reflection is considerably easier compared to the other methods and, perhaps more interestingly, the method convergences at a rate of $\mathcal{O}(N^{-1/2})$ regardless of the dimension of the integral. Its main disadvantage is probably the statistical nature of the convergence, although Veach [8] noted that this property can be favorably used by calculating the variance of the sample set and estimate the quality of the solution. Due to all of those factors, the method used in this project was Monte Carlo. A more detailed description of each methodology and the motivations for each different approach can be found in standard textbooks such as [1,2]. The accuracy of aerothermodynamic simulations is also strongly related to the chosen physico-chemical models, particularly, in the thermodynamic nonequilibrium or non-LTE region [9]. As such, it is crucial to have available experimental data to validate the simulations. Among the several experiments flown up to the present, one of the most important for space applications (our main interest) is certainly the FIRE II mission [10]. The project was developed in the 60s to gather data of the aerothermal environment surrounding a non-ablative capsule during an interplanetary re-entry. The experiment provided spectroscopic data and heat flux measurements spanning various aerothermodynamic regimes. As such, the conditions of one trajectory point of FIRE II were replicated with numerical simulations in order to assess the quality of the Monte Carlo solver and the physical models used.